Analyzing Multi-Key Security Degradation

Abstract. The multi-key, or multi-user, setting challenges cryptographic algorithms to maintain high levels of security when used with many different keys, by many different users. Its significance lies in the fact that in the real world, cryptography is rarely used with a single key in isolation. A folklore result, proved by Bellare, Boldyreva, and Micali for public-key encryption in EUROCRYPT 2000, states that the success probability in attacking any one of many independently keyed algorithms can be bounded by the success probability of attacking a single instance of the algorithm, multiplied by the number of keys present. Although sufficient for settings in which not many keys are used, once cryptographic algorithms are used on an internet-wide scale, as is the case with TLS, the effect of multiplying by the number of keys can drastically erode security claims. We establish a sufficient condition on cryptographic schemes and security games under which multi-key degradation is avoided. As illustrative examples, we discuss how AES and GCM behave in the multi-key setting, and prove that GCM, as a mode, does not have multi-key degradation. Our analysis allows limits on the amount of data that can be processed per key by GCM to be significantly increased. This leads directly to improved security for GCM as deployed in TLS on the Internet today.

1 Introduction

A crucial aspect to analyzing cryptographic algorithms is modeling real-world settings. These models should not only accurately reflect the limits imposed by the environments and the security properties desired, but they should also produce meaningful ways to estimate how security deteriorates with use. In particular, in practice, algorithms are fixed, and hence so are key sizes, block sizes, groups, and various other parameters. Therefore it is important to be able to compute adversarial success probabilities relative to their resources as precisely as possible.

For example, block ciphers have traditionally been analyzed in a setting where adversaries are given access to the encryption and decryption oracles keyed with a value chosen uniformly at random, unknown to the adversary. For many purposes using a block cipher which is secure in this model is sufficient, barring easy access to side channel information. Estimates for adversarial success are obtained by analyzing the best known attacks against the block cipher, relative to both *computational* complexity, or the cost of running the attack as measured according to, say, time and memory, and *data* complexity, or the amount of data the adversary receives from the oracles, measured in, for example, bits. Taking a concrete example, AES [22], one can map the cost needed to recover a key, as

is done in Fig. 1a. For the 128-bit key version of full round AES, the best known attacks have computational complexity improving over brute-force search by a factor of 2 to 4, and arbitrarily increasing data complexity does not allow one to reduce computational complexity much.

The analysis of block ciphers contrasts sharply with that of *modes of operation* for block ciphers, which are algorithms that repeatedly use block cipher calls to achieve security properties beyond what a block cipher can provide on its own. As an important (but by no means the only) example, the Authenticated Encryption with Associated Data (AEAD) [47] mode of operation GCM [37] uses a block cipher to achieve data confidentiality and authenticity simultaneously, formalized in a setting where adversaries are given access to keyed encryption and decryption oracles. The security of GCM is proved by showing that any AEAD adversary against the mode can be converted into an adversary against the pseudo-randomness of the underlying block cipher [32, 42]. Thus, if GCM were to be used with AES, then AES-GCM is secure in the AEAD sense under the assumption that AES is secure as a pseudo-random permutation (PRP).

However, the quality of the reduction from AES to AES-GCM deteriorates with use. Following the concrete security paradigm [7], this degradation has been quantified to be roughly $\sigma^2/2^{128}$ [32, 42], where σ is the number of blocks of ciphertext seen by the adversary, or its data complexity. This is depicted in Fig. 1b. Therefore, quantifying AES-GCM's security relies not only on understanding AES's security, but also on how GCM as a mode degrades security.

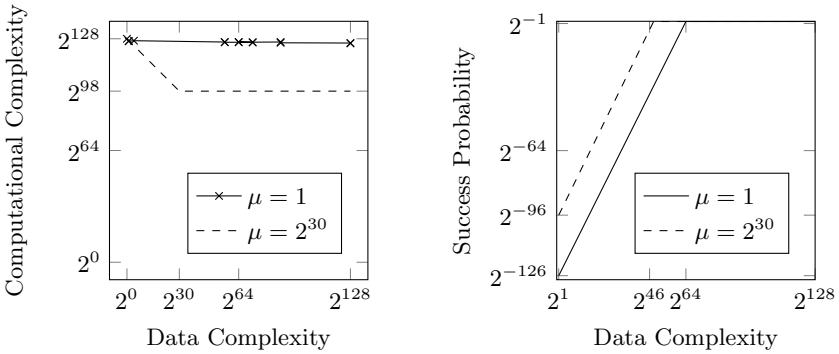
Note that in the case of AES-GCM, an understanding of how adversarial computational and data complexity affect security can be built by looking at AES and GCM separately. AES's security degrades as computational resources increase, but increased data complexity does not seem to introduce better attacks. GCM's security as a mode degrades as data complexity increases, but computational complexity does not play a role.

1.1 From Single-Key to Multi-Key

The security models described earlier for block ciphers and modes gave the adversaries access to encryption and decryption oracles operating under a single key. However, in practice cryptographic algorithms are used by many different users, each potentially with many different keys. For example, AES-GCM is now widely used in TLS to protect web traffic via HTTPS,[1] and is currently used by millions, or perhaps billions, of users daily. Hence it is important to understand what happens to security in the so-called *multi-key* setting, where adversaries are successful if they compromise the security of one out of many users, meaning their winning condition is a disjunction of single key winning conditions.

For block ciphers the picture changes both quantitatively and qualitatively. Whereas in the single-key setting, the best attacks against AES do not improve with increased data complexity, in the multi-key setting they do, as depicted in

[1] The latest figures from the ICSI Certificate Notary (`https://notary.icsi.berkeley.edu/`) suggest that more than 70% of all TLS connections use AES-GCM.



(a) Key recovery attacks against full AES-128. All attacks have success probability one. Data on single-key attacks from [17, 18, 50].

(b) Upper bound on attack success probability against the mode GCM, based on the equation $\mu\sigma^2/2^{128}$, where σ is the data complexity.

Fig. 1: Comparison of how data complexity affects attacks against AES-128 and GCM in the single ($\mu = 1$) and multi-key ($\mu = 2^{30}$) settings. Note that the AES graph depicts attacks, whereas the GCM graph depicts upper bounds on attack success probability.

Fig. 1a. As observed first by Biham [14], and later refined as a time-memory-data trade-off by Biryukov, Mukhopadhyay, and Sarkar [15], one can take advantage of the fact that recovering a key out of a large group of keys is much easier than targeting one key. The same observation can be applied to any deterministic symmetric-key algorithm, as is done for MACs by Chatterjee, Menezes, and Sarkar [21].

More generally, a folklore result guarantees that the attack success probability increases by at most a factor μ when moving from the single-key to the multi-key setting with μ keys. In the case of key recovery against AES, the fact that this increase is necessary can be illustrated with an actual attack. For the mode GCM, a security bound involving a factor μ is easily established using a hybrid argument, meaning that the adversarial success probability is bounded by roughly $\mu \cdot \sigma^2/2^{128}$. Bellare and Tackmann [11] were the first to formalize authenticated encryption in the multi-key setting and to analyze countermeasures against multi-key attacks in the context of TLS 1.3. Their work similarly establishes bounds containing a μ-factor. This leads to a significant security degradation when there are many GCM instances present, as illustrated in Fig. 1b. Unfortunately, this is exactly the situation faced in large-scale deployments of AES-GCM such as TLS.

Unlike block ciphers, there are no known attacks which establish the tightness of the $\mu \cdot \sigma^2/2^{128}$ security bound for the GCM mode. Assuming there were such an attack, then the bound would say that, using the *same* amount of resources σ as a single-key adversary, a multi-key adversary would be able to *increase* its

success probability by a factor of μ. Therefore a successful multi-key adversary against the GCM mode would be able to use its resources much more efficiently than a single-key attacker would.

Quantifying this difference, in order for a single-key adversary to be able to achieve the same bound $\mu\sigma^2/2^{128}$ using σ_1 resources, $\sigma_1^2/2^{128}$ must equal $\mu\sigma^2/2^{128}$, or in other words, $\sigma_1 = \mu^{1/2}\sigma$. In particular, $\sigma/\mu = \mu^{-3/2}\sigma_1$, and so a multi-key adversary's per-key cost would decrease proportional to $\mu^{-3/2}$ relative to a single-key adversary's per-key cost, while achieving the same success probability. So, if there were a multi-key adversary interacting with, say, ten thousand GCM instances, and matching the generic bound, then in order for a single-key adversary to match the multi-key adversary's success probability, it must spend a factor of one million more than a multi-key adversary has to spend per key. Note that even in the case of AES, the best known multi-key attack does not make better use of its data resources: it achieves the same success probability as the single-key attacks with roughly the same per-key data cost, namely, one plaintext-ciphertext pair per key.

1.2 Overview and Contributions

We set out to understand why there are seemingly no attacks matching the multi-key bounds established by the folklore result, and by formal proofs in certain cases, against modes such as GCM. To do so, we systematically analyze the transition from games in which adversaries are given access to oracles representing a single, keyed algorithm, to games where adversaries are given access to multiple oracles representing different, independently keyed instances of an algorithm.

The fact that the folklore result is the best generic reduction possible has already been established by Bellare, Boldyreva, and Micali [4], where they construct a public-key encryption scheme which necessarily has the μ-degradation. However, we take the informal guidance provided by such special cases a step further in Sect. 2, and point out that the multi-key setting is the natural one in which to consider *weak keys*, by illustrating how they can allow multi-key adversaries to make better use of resources in comparison with single-key adversaries.

In Sect. 3 we continue by distilling a sufficient condition under which adversaries gain no advantage in the multi-key setting over the single-key setting. Informally, the condition states that it should always be better to attack an instance of an algorithm for which the adversary is given more information, as measured by the number of queries made to the instance. Note that this condition is not satisfied for algorithms with weak keys: if the adversary knows that an instance uses a strong key, then it might be better for it to take its chances with an instance for which it has little information, but where it might get to attack a weak key.

Although intuitively appealing, the condition that we extract can be difficult to use as a criterion in security analyses. Therefore in Sect. 4 we compare various methods for proving the absence of multi-key degradation, such as for the PRP-

PRF switch and for Wegman-Carter MACs [52]. Finally, we prove that GCM has security bounds that are independent of μ using our sufficient condition.

1.3 Interpretation

Our claim that GCM enjoys a multi-key security bound that does not depend on μ might seem counter-intuitive. After all, GCM uses a block cipher, and, as illustrated above with an attack, all block ciphers necessarily have security that degrades with μ. It seems natural that one can apply a similar attack to GCM thereby establishing μ-degradation.

The result concerning GCM is a statement made once the underlying block cipher is replaced by a uniformly distributed random permutation, which is a standard technique used to reduce the block cipher's insecurity to GCM's insecurity when used with that block cipher. Stated as an imprecise formula, for a single key, we have that

$$\mathsf{Insecurity}(GCM, E) \leq \mathsf{Insecurity}(GCM, \pi) + \mathsf{Insecurity}(E)\,, \tag{1}$$

where E is the keyed block cipher, and π is the random permutation. Passing to the multi-key setting means that one now considers the insecurity of GCM with multiple independently keyed block ciphers $E_{K_1}, E_{K_2}, \ldots, E_{K_\mu}$, which are then replaced with independent uniformly distributed random permutations $\pi_1, \pi_2, \ldots, \pi_\mu$. Saying that GCM as a mode does not degrade with μ is a statement about how the insecurity of GCM with $\pi_1, \pi_2, \ldots, \pi_\mu$ does not degrade as the number of independent permutations increases, and as a result, the reduction from the insecurity of the underlying block ciphers $E_{K_1}, \ldots, E_{K_\mu}$ to the insecurity of $(GCM, E_{K_1}, \ldots, E_{K_\mu})$ does not deteriorate according to μ. However, any multi-key attack against E still holds, and is taken into account when considering the term corresponding to the insecurity of E in the multi-key version of (1).

In other words, what we are able to show is that security does not degrade "doubly", once for GCM and once again for the block cipher, when the number of keys increases. More importantly, one can conclude that in order to understand the multi-key security of AES-GCM, one can focus on the multi-key security of AES.

1.4 Practical Implications

This insight has an immediate and important practical consequence. Recently the TLS Working Group of the IETF has been considering data limits for the AEAD schemes to be used in TLS 1.3, the new version of TLS under development. Amongst these schemes is AES-GCM. Luykx and Paterson provided an analysis of the safe data limits for AES-GCM.[2] They did this by first analyzing the known

[2] See `https://mailarchive.ietf.org/arch/msg/tls/M-fcRtoeCtMxDNtMsPrUsBV5rgk`.

bounds for AES-GCM in the single-key setting and then applying a factor μ in order to obtain bounds for the multi-key setting. The safe data limits for AES-GCM turned out to be surprisingly small, especially in the multi-key case: the current draft of TLS 1.3 states that, in the single-key setting, only $2^{24.5}$ full-size records may be encrypted on a given connection while keeping a safety margin of approximately 2^{-57}. Following the analysis of Luykx and Paterson, one would infer that the safety margin decreases proportionately with μ in the multi-key case. This analysis prompted the TLS Working Group to mandate a key updating mechanism for TLS 1.3. Our multi-key analysis for AES-GCM shows that this additional feature, which adds complexity to an already complex protocol, may be unnecessary.

1.5 Other Work Reducing Multi-Key Degradation

The approach outlined above is that of the *standard model*. Bellare and Tackmann [11] use the *ideal cipher model* in order to understand how different modifications to GCM improve resistance against key recovery in the multi-key setting. Their goal is not to establish μ-independence, but to rather extend the effective key length of GCM over that of the underlying block cipher in order to make key recovery more difficult. However, for GCM, they end up with a factor of μ in their security bounds as a consequence of their method of analysis, whereas our results show that this is not inevitable.

In special cases the dependence on μ disappears. Bellare, Bernstein, and Tessaro show that this is the case with AMAC [3]. Hoang and Tessaro (HT) [29] establish a similar result for key-alternating ciphers, and even show more generally that if a construction has transcripts satisfying some special properties, then μ no longer appears when considering bounds on indistinguishability. The HT-condition is a useful sufficient condition because it only places a requirement on how an upper bound on the difference between the probabilities of two transcripts behaves. However, its applicability is limited, as we will illustrate in Sect. 4.4, because it does not provide a meaningful bound when considering integrity. In concurrent work, Hoang and Tessaro [30] generalize their previous approach, and apply it to double encryption. Their transcript-driven approach provides different insight into how to prove the lack of multi-key security degradation, and can be applied equally well to GCM to arrive at the same conclusion as we do.

1.6 Further Related Work

A significant amount of work has gone into understanding what happens when security properties are analyzed in the multi-key setting in a variety of different contexts. These include public key encryption [5], key establishment protocols [9,16], signatures [38], message authentication codes [3,21], tweakable block ciphers [27,54], and hybrid encryption [20,53]. Bader et al. recently established impossibility results showing that a loss of a factor μ is inevitable when moving to the multi-key setting for a range of public-key primitives [1]. Most recently,

Shrimpton and Terashima [49] introduced a new model in order to bridge gaps between standard and ideal model bounds to analyze settings where the standard model bounds provide little assurance of security, like the multi-key setting. Other research on security of block ciphers in the multi-key setting includes the works by Mouha and Luykx [39], Tessaro [51], and Fouque et al. [24]. However, there is no systematic treatment of the problem like that provided in our work.

2 Weak Key Attacks

Bellare, Boldyreva, and Micali (BBM) [4] give an example of a public-key encryption scheme which illustrates that the factor μ is necessary in any generic bound. The example creates a new public-key encryption scheme from an existing one by introducing a "bad" event into the construction which occurs with some fixed probability and allows adversaries to win easily. When interacting with a single instance, the bad event occurs with low probability. However, by working with multiple instances, one can increase the chances of triggering the bad event.

The BBM example illustrates a type of attack one can perform against algorithms in the multi-key setting that is different from the time-memory-data trade-off applied to AES [15]. The multi-key attack against AES precomputes the encryption of a plaintext under a large set of keys, and hopes for a collision between the precomputed values and the oracles in order to immediately recover keys. This attack can be applied to any block cipher, no matter how secure it is.

An analogue of the BBM example in the block cipher setting is a block cipher with *weak keys*, these being keys under which one can attack the block cipher much more efficiently than expected. For example, the recently introduced block cipher Midori64 [2] has a class of 2^{32} weak keys [26] out of 2^{128}, which when identified (which can be done with a single query), can lead to key recovery within computational complexity 2^{16} and data complexity 2. When analyzed in the single-key setting, attackers either get a strong key, in which case key recovery presumably still takes roughly $2^{128} - 2^{32}$ time, or a weak key, leading to a speed-up. When analyzed in the multi-key setting, the chances of finding a weak key are much higher, and adversaries can allocate their resources more efficiently.

A good strategy for a multi-key adversary attacking an algorithm with weak keys would be to first spend some resources across its μ oracles to detect if one of them is using a weak key, and then to allocate as many resources as necessary to attack the weak key. If P is the probability that a key is weak, C_W the cost to break the algorithm with probability one given that it is using a weak key, and C_D the cost to detect a weak key, then in cost at most $C_W + \mu C_D$, the success probability of breaking the algorithm can be improved by a factor

$$\frac{1-(1-P)^{\mu}}{P} = 1 + (1-P) + (1-P)^2 + \cdots + (1-P)^{\mu-1}\,, \tag{2}$$

which is the probability of finding at least one weak key out of μ over the probability of a single key being weak. If P is small, then this means the success

probability increases by a factor almost linear in μ. Plugging in the numbers for Midori64, we have that a multi-key adversary interacting with $\mu = 2^{16}$ keys, with computational complexity 2^{17} and data complexity $\mu + 2$ has success probability a factor of approximately 2^{16} better than the single-key attack, which has computational complexity 2^{16} and data complexity 2.

When formally analyzing modes of operation, time-memory-data key recovery attacks are usually taken out of consideration because the block cipher is replaced with a uniformly random permutation. Instead, attacks that might improve in the multi-key setting are those that take advantage of bad events in security proofs.

3 When Multiple Oracles Do Not Benefit Adversaries

In this section we introduce and prove the sufficient condition characterizing when adversaries have no advantage with multiple oracles over a single oracle. We start by introducing basic notation and definitions used throughout the section, and then review the generic folklore bound. We end the section by showing how the condition is sufficient.

3.1 Notation

Given a set X, $\mathsf{X}^{\leq q}$ denotes the set of non-empty sequences of X of length less than or equal to q, and X^+ denotes the set of non-empty arbitrary length sequences of elements of X. Given $\boldsymbol{x} \in \mathsf{X}^+$, $|\boldsymbol{x}|$ denotes its length, and $\lfloor \boldsymbol{x} \rfloor_q$ denotes the first q elements of $\boldsymbol{x}$, that is, $(x_1, \ldots, x_q)$, and all of $\boldsymbol{x}$ if $q \geq |\boldsymbol{x}|$. If $\mathsf{W} \subset \mathsf{X}^+$, then $\lfloor \mathsf{W} \rfloor_q$ consists of $\lfloor \boldsymbol{x} \rfloor_q$ for $\boldsymbol{x} \in \mathsf{W}$. The concatenation of two sequences $\boldsymbol{x}, \boldsymbol{x}' \in \mathsf{X}^+$ is denoted $\boldsymbol{x} \| \boldsymbol{x}'$.

A prefix of a sequence $\boldsymbol{x}$ is a sequence $\boldsymbol{x}'$ where $\boldsymbol{x}' = \lfloor \boldsymbol{x} \rfloor_i$ for some $i \leq |\boldsymbol{x}|$. An extension of a sequence $\boldsymbol{x}$ is a sequence $\boldsymbol{x}'$ such that $\boldsymbol{x}$ is a prefix of $\boldsymbol{x}'$.

3.2 Games and Adversaries

We use Maurer's random systems formalization [35, 36] with slightly different notation.

A game G from X to Y is a tuple $(\mathbf{O}, \mathsf{W})$ consisting of an (X, Y)-system $\mathbf{O}$, meaning $\mathbf{O}$ accepts inputs from X and generates outputs in Y which can depend probabilistically on the current input and all previous outputs, and a random variable $\mathsf{W} \subset (\mathsf{X} \times \mathsf{Y})^+$ which may depend on $\mathbf{O}$, representing the "winning" transcripts. Our formalization of a game G can be viewed as an $(\mathsf{X}, \mathsf{Y} \times \{0,1\})$ random system in Maurer's formalization by concatenating the oracle $\mathbf{O}$ with a random system that outputs 1 if the current transcript is in W. We write $\mathbf{O}\langle \boldsymbol{t} \rangle$ to mean the event that

$$(\mathbf{O}(x_1), \mathbf{O}(x_2), \ldots, \mathbf{O}(x_\ell)) = (y_1, y_2, \ldots, y_\ell)\,, \tag{3}$$

where $\boldsymbol{t} = ((x_1, y_1), \ldots, (x_\ell, y_\ell))$. Note that the order of the queries in the transcript is important since $\mathbf{O}$ could be stateful.

An adversary $\mathbf{A}$ interacting with $G = (\mathbf{O}, \mathsf{W})$ is a (Y, X)-system, which produces a sequence of inputs $(x_1, x_2, \ldots) \in \mathsf{X}^+$, where x_i is generated using $y_1, y_2, \ldots, y_{i-1}$ with $y_j = \mathbf{O}(x_j)$ for $j = 1, \ldots, i-1$; note that x_1 is generated without any $\mathbf{O}$-output. We let $\mathbf{A}^{\mathbf{O}}$ denote the sequence $((x_1, y_1), (x_2, y_2), \ldots) \in (\mathsf{X} \times \mathsf{Y})^+$, which is a random variable. We say that a transcript $\mathbf{A}^{\mathbf{O}}$ wins if $\mathbf{A}^{\mathbf{O}} \in \mathsf{W}$, and write $\mathbf{A}\langle \boldsymbol{t} \rangle$ for $\boldsymbol{t} = ((x_1, y_1), (x_2, y_2), \ldots, (x_q, y_q)) \in (\mathsf{X} \times \mathsf{Y})^+$ to denote the event that $\mathbf{A}$ produces x_i as the ith oracle input when given $(y_1, y_2, \ldots, y_{i-1})$ as oracle outputs, for $i = 1, \ldots, q$.

Let q be a non-negative integer, then the advantage of an adversary $\mathbf{A}$ winning game G within q queries is

$$\operatorname*{adv}_{G,q} \mathbf{A} := \mathbb{P}\left[\mathbf{A}^{\mathbf{O}} \in \lfloor \mathsf{W} \rfloor_q\right] . \tag{4}$$

Ultimately, the quantity we are interested in is

$$\sup_{\mathbf{A}} \operatorname*{adv}_{G,q} \mathbf{A} . \tag{5}$$

Without loss of generality, we may focus on deterministic adversaries, since for all $\mathbf{A}$,

$$\begin{aligned} \operatorname*{adv}_{G,q} \mathbf{A} &= \mathbb{P}\left[\mathbf{A}^{\mathbf{O}} \in \lfloor \mathsf{W} \rfloor_q\right] && (6)\\ &= \sum_{A \in \mathbb{D}} \mathbb{P}\left[\mathbf{A}^{\mathbf{O}} \in \lfloor \mathsf{W} \rfloor_q \;\middle|\; \mathbf{A} = A\right] \cdot \mathbb{P}\left[\mathbf{A} = A\right] && (7)\\ &\leq \sup_{A \in \mathbb{D}} \mathbb{P}\left[A^{\mathbf{O}} \in \lfloor \mathsf{W} \rfloor_q\right] , && (8) \end{aligned}$$

where $\mathbb{D}$ represents all deterministic adversaries. Furthermore, we generally assume that the input and output spaces of our oracles are finite. This means there are finitely many optimal choices for adversaries to make, hence the above supremum is attained, and can be described as a maximum. For this reason we can speak of *optimal* adversaries, that is, any adversary that attains the maximum advantage given a particular oracle, game, and query bound.

Unless specified otherwise, we only consider games that are *monotone*, that is, $\boldsymbol{t} \in \mathsf{W}$ implies that all extensions $\boldsymbol{t}'$ of $\boldsymbol{t}$ are in W. In monotone games it is also useful to consider the first query which triggers the winning event: before this query is made the adversary has not yet won, and this is the first query for which one can say that the adversary has won.

3.3 Multi-Oracle Games and an Existing Bound

Consider an adversary $\mathbf{A}$ interacting with multiple independent games

$$\{G_i = (\mathbf{O}_i, \mathsf{W}_i)\}_{i \in I} , \tag{9}$$

with as goal to win the disjunction of the G_i. Letting X_i denote the domain of oracle $\mathbf{O}_i$ and X the set of elements (i, x) such that $x \in \mathsf{X}_i$, the game $G = (\mathbf{O}, \mathsf{W})$ that $\mathbf{A}$ plays can be defined with the single oracle $\mathbf{O}(i, x) = \mathbf{O}_i(x)$, and by W where $\boldsymbol{t} \in \mathsf{W}$ if the projection $\Pi_i \boldsymbol{t}$ of $\boldsymbol{t}$ onto the $\mathbf{O}_i$-queries is in W_i for some i.

If we know the security bounds for each G_i, then there is a simple way of bounding $\mathbf{A}$'s advantage without computing it from scratch: for each $i \in I$ construct an adversary $\mathbf{A}_i$ which runs $\mathbf{A}$, plays game G_i, and simulates all the other games independently. The adversary $\mathbf{A}_i$ perfectly simulates $\mathbf{A}$'s game precisely because game G_i is independent of all other games. Moreover, $\mathbf{A}_i$ wins if $\mathbf{A}$ does in game G_i. Then, by a simple averaging argument over a random choice of i, $\mathbf{A}$'s advantage within q queries can be bounded by the sum of the advantages of the $\mathbf{A}_i$ for $i \in I$, or

$$\operatorname*{adv}_{G,q} \mathbf{A} \leq \sum_{i \in I} \operatorname*{adv}_{G_i,q} \mathbf{A}_i \,. \tag{10}$$

The setting we focus on is when the G_i are independent instances of the same game G_1. Given a game $G = (\mathbf{O}, \mathsf{W})$ from X to Y, define $\overline{G} = (\overline{\mathbf{O}}, \overline{\mathsf{W}})$ to be the game giving access to the family $\{\mathbf{O}_i\}_{\mathbb{N}}$, which is a family of independently distributed copies of $\mathbf{O}$ indexed by $\mathbb{N}$, and where $\boldsymbol{t} \in \overline{\mathsf{W}}$ if $\Pi_i \boldsymbol{t} \in \mathsf{W}$ for some $i \in \mathbb{N}$. In this case the generic multi-key bound simplifies to

$$\operatorname*{adv}_{G,q} \mathbf{A} \leq \mu \cdot \operatorname*{adv}_{G_1,q} \mathbf{A}_1 \,, \tag{11}$$

where μ is the size of I, or a bound on the number of different oracles that $\mathbf{A}$ queries. This bound can be applied to any game, and has been in the case of public-key encryption [5] and PRFs [3, 6].

Definition 1. *The oracle* $\mathbf{O}$ *does not exhibit multi-key security degradation with respect to* $G = (\mathbf{O}, \mathsf{W})$*, if for all* $q > 0$

$$\sup_{\mathbf{A}} \operatorname*{adv}_{\overline{G},q} \mathbf{A} \leq \sup_{\mathbf{A}} \operatorname*{adv}_{G,q} \mathbf{A} \,. \tag{12}$$

3.4 Sufficient Condition

Since the goal of multi-oracle adversaries is to win any of the single-oracle games it is given, finding the optimal strategy is a question of targeting those single-oracle games for which it has the highest chance of winning, relative to its query allotment. The information that the adversary can work with is the transcripts produced from each single-oracle game and how many queries it has left. So, for example, a good strategy for an adversary might be to query each oracle once, and to estimate based on all of the transcripts which oracle is the weakest, and then to focus on the weakest one.

Conversely, if all of the oracles are equally strong, then, intuitively, one might think that it does not make a difference that the adversary can work with more than one oracle, since there is little difference between the various oracles, and

the adversary's best strategy would seem to be to focus its effort on just one of them. However, to formally establish this we require an additional condition: it must be the case that when an optimal single-oracle adversary is given additional knowledge about the oracle, then its chance of winning the game does not *decrease* relative to an optimal single-oracle adversary given less knowledge. Now, if an adversary is interacting with multiple oracles, and it has more information about one oracle over the others, then its best strategy is to stick to that oracle instead of switching to another one.

This condition breaks down, for instance, if a construction has weak keys: if an adversary has the knowledge that its oracle is using a weak key, then it might have better advantage in winning the game versus an oracle where there is still a chance of interacting with a strong key.

Below, we formalize the idea of giving adversaries additional knowledge via *games with advice*, which is equivalent to the concept of projected systems and their advantage by Gaži and Maurer [25]. Gaži and Maurer's projected systems explicitly define new conditional probability distributions which explain the behavior of the system from a given starting transcript. For our purposes we do not need to use the definition of a projected system directly, only the associated advantage definition.

Definition 2. *Let $G = (\mathbf{O}, \mathsf{W})$ be a game and $\boldsymbol{t} \in (\mathsf{X} \times \mathsf{Y})^+$ be a transcript. Then G with advice $\boldsymbol{t}$, denoted $G^{\boldsymbol{t}}$, is defined as $(\mathbf{O}, \mathsf{W}^{\boldsymbol{t}})$, where $\boldsymbol{s} \in \mathsf{W}^{\boldsymbol{t}}$ if and only if $\boldsymbol{t}$ is a prefix of $\boldsymbol{s}$ and $\boldsymbol{s} \in \mathsf{W}$. The advantage of adversary $\mathbf{A}$ in winning game $G^{\boldsymbol{t}} = (\mathbf{O}, \mathsf{W}^{\boldsymbol{t}})$ within q queries is*

$$\operatorname*{adv}_{G^{\boldsymbol{t}},q} \mathbf{A} := \mathbb{P}\left[\mathbf{A}^{\mathbf{O}} \in \lfloor \mathsf{W}^{\boldsymbol{t}} \rfloor_q \;\middle|\; \mathbf{O}\langle \boldsymbol{t} \rangle \;, \boldsymbol{t} \notin \mathsf{W}\right] . \tag{13}$$

The definition below contains the additional condition we need in order to show in Thm. 1 that multi-oracle adversaries do not gain any advantage relative to single-oracle adversaries. Note that it only looks at single-oracle adversaries, meaning if a game satisfies the condition, then one can conclude something about multi-oracle adversaries just by looking at single-oracle adversaries.

Informally, the condition states the following. Take a game G, a transcript $\boldsymbol{t}$, and *any* shorter transcript $\boldsymbol{t}'$ — it does not have to be a prefix of $\boldsymbol{t}$. Then two settings are compared: one in which adversaries are given $\boldsymbol{t}$ as starting information, and one in which adversaries are given $\boldsymbol{t}'$ as starting information. In both settings adversaries are allotted the same number of queries left to make, computed as $q - |\boldsymbol{t}|$ in the condition. Then the condition states that optimal adversaries starting with $\boldsymbol{t}$ should have advantage greater than or equal to optimal adversaries starting with $\boldsymbol{t}'$, and this should hold for all $\boldsymbol{t}$ which can result from the interaction between an optimal adversary and the game, and all $\boldsymbol{t}'$ shorter than $\boldsymbol{t}$. Even though the condition might seem strong, the proof of Thm. 1 is non-trivial. In Lem. 3 we show that GCM's underlying polynomial hash satisfies it.

The other details in the condition are there to remove pathological situations, for example removing transcripts $\boldsymbol{t}$ which could never occur, or to remove situations that do not need to be taken into account in the condition in order for the

proof to hold, for example removing transcripts $\boldsymbol{t}$ and $\boldsymbol{t}'$ for which adversaries are guaranteed to win. For this purpose, define transcript $\boldsymbol{t}$ to be $(\mathbf{A}, G)$-*meaningful* if

$$\mathbb{P}\left[\lfloor \mathbf{A}^{\mathbf{O}} \rfloor_{|\boldsymbol{t}|} = \boldsymbol{t}, \boldsymbol{t} \notin \mathsf{W}\right] > 0\,. \tag{14}$$

Definition 3 (Progressive Games). *Let $G = (\mathbf{O}, \mathsf{W})$ be a monotone game from X to Y and Q be any non-negative integer. Suppose that for all $q \leq Q$, all optimal adversaries $\mathbf{A}$, all $(\mathbf{A}, G)$-meaningful $\boldsymbol{t}$ such that $q' := q - |\boldsymbol{t}| \geq 0$, we have that, for all transcripts $\boldsymbol{t}'$ with $|\boldsymbol{t}'| < |\boldsymbol{t}|$ that are meaningful with respect to some adversary,*

$$\sup_{\mathbf{C}} \operatorname*{adv}_{G^{\boldsymbol{t}}, q} \mathbf{C} \geq \sup_{\mathbf{B}} \operatorname*{adv}_{G^{\boldsymbol{t}'}, q'+|\boldsymbol{t}'|} \mathbf{B}\,. \tag{15}$$

Then G is said to be progressive.

Theorem 1. *Let $\mathbf{O}$ be an oracle and $G = (\mathbf{O}, \mathsf{W})$ be a progressive game. Then $\mathbf{O}$ does not exhibit multi-key security degradation.*

3.5 Proof of Theorem 1

Notation. Let $[0, 1]$ be the unit interval, and let $\cdot$ denote the dot product of two equal-length elements of $[0, 1]^+$, i.e.

$$\boldsymbol{x} \cdot \boldsymbol{y} = \sum_i x_i y_i\,. \tag{16}$$

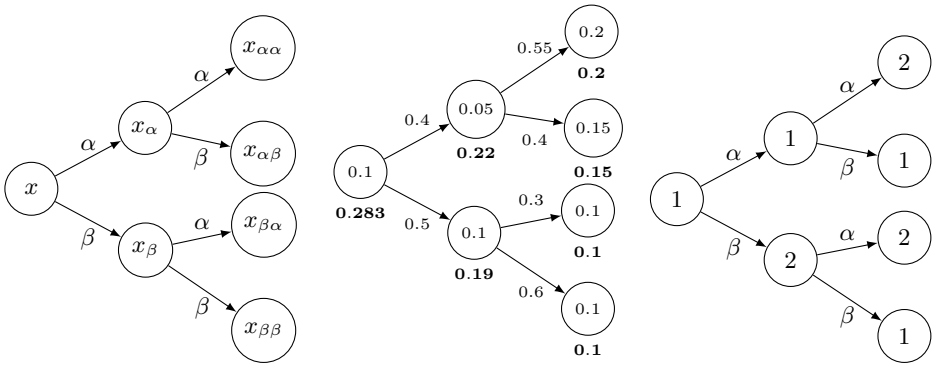
Let $\boldsymbol{x} \in [0, 1]^+$, then $\mathbf{1} \cdot \boldsymbol{x}$ denotes the dot product of $\boldsymbol{x}$ with a vector consisting of $|\boldsymbol{x}|$ ones (or put simply, $\mathbf{1} \cdot \boldsymbol{x}$ is the sum of the components in $\boldsymbol{x}$).

Decision Trees. The interaction between a game G and a deterministic adversary $\mathbf{A}$ can be viewed as a *decision tree* as follows. The adversary produces a first input $x_1 \in \mathsf{X}$ to the oracle $\mathbf{O}$, which represents the root of the tree. The oracle produces an output $y_1 \in \mathsf{Y}$, and depending upon the output, $\mathbf{A}$ decides its next oracle input. Each of the possible oracle outputs $y_1 \in \mathsf{Y}$ results in an edge extending from the root to a child node, which contains $\mathbf{A}$'s second oracle query, assuming (x_1, y_1) has occurred. Then, starting from a child node, we extend the tree further by adding edges according to the second oracle output, connecting them to the third oracle inputs. Without loss of generality, we may restrict ourselves to decision trees where each edge has a non-zero chance of occurring: if the output y_1 is not possible with input x_1, then we do not include that edge in the tree.

Consider for example some adversary $\mathbf{A}_H$ playing a game $H = (\mathbf{R}, \mathsf{V})$ where the oracle $\mathbf{R}$'s output domain is $\{\alpha, \beta\}$. Then the root of $\mathbf{A}_H$'s decision tree will contain some value x representing an input to $\mathbf{R}$, and is connected by two edges, labeled by α and β respectively, to two child nodes. The child node connected to x via α represents the adversary's second oracle input assuming the first oracle output was α, and similarly for the other child node. Fig. 2a illustrates what

the tree looks like for this example with deterministic adversaries making three queries. Throughout this section we use the notation $\mathbf{A}_H$ and $H = (\mathbf{R}, \mathsf{V})$ to refer to this running example, and the notation $\mathbf{A}$ and $G = (\mathbf{O}, \mathsf{W})$ to refer to a generic adversary and game.

The *level* of the root node equals one, and a child of a node with level ℓ has level $\ell + 1$. Each node in the tree is connected by a unique path to the root. Let x_i be a node with path $x_1 \xrightarrow{y_1} x_2 \xrightarrow{y_2} \cdots x_{i-1} \xrightarrow{y_{i-1}} x_i$ connecting it to the root. Then the transcript associated to the node x_i is $((x_1, y_1), (x_2, y_2), \ldots, (x_{i-1}, y_{i-1}))$.



(a) Decision tree of $\mathbf{A}_H$. Each edge is labeled by an oracle output, and each node is labeled by an oracle input.

(b) Probability label P_T for tree depicted left. Written in bold underneath each node is its value ν.

(c) Decision tree of a multi-oracle adversary $A_{\overline{H}}$ making queries to two different oracles.

Fig. 2: An example of how a decision tree is constructed (left) along with a possible probability labeling (center) from the game $H = (\mathbf{R}, \mathsf{V})$ with adversary $\mathbf{A}_H$, as well the decisions made by a possible multi-oracle adversary $\mathbf{A}_{\overline{H}}$.

Probability Labeling. Starting from a decision tree T for adversary $\mathbf{A}$ and game G, we construct a labeling P_T consisting of probabilities from which one can compute the adversary's advantage. The root node in T is labeled with the probability that the adversary wins on the first query. If y denotes the label of an edge emanating from the root node in T, then the corresponding label in P_T is the probability that the first query does not win, and the output of the first query is y. The node at the end of this edge is then labeled by the probability that the second query wins, given that the first query does not win and the output of the first query is y. Note that the sum of the label of the root node and all its edges must equal one, since either the first query wins, or the first query does not win, and the edges split up the event that the first query does not win according to the output of the first query.

The labeling P_T is then extended to the entire tree T using a similar process. Given a node x_i and its associated transcript $\boldsymbol{t}$, the node x_i is labeled by the probability that the ith query x_i wins given that the preceding transcript $\boldsymbol{t}$ does not win, i.e. $\boldsymbol{t} \notin \mathsf{W}$, and $\boldsymbol{t}$ has occurred, i.e. $\lfloor \mathbf{A}^{\mathbf{O}} \rfloor_{|\boldsymbol{t}|} = \boldsymbol{t}$, or in other words, letting $P_T(x_i)$ denote the labeling of node x_i,

$$P_T(x_i) := \mathbb{P}\left[\lfloor \mathbf{A}^{\mathbf{O}} \rfloor_i \in \mathsf{W} \;\middle|\; \lfloor \mathbf{A}^{\mathbf{O}} \rfloor_{i-1} = \boldsymbol{t}, \boldsymbol{t} \notin \mathsf{W} \right]. \tag{17}$$

In the same way, an edge $x_i \xrightarrow{y_i} x_{i+1}$ is labeled in P_T by

$$P_T(x_i \xrightarrow{y_i} x_{i+1}) := \mathbb{P}\left[\lfloor \mathbf{A}^{\mathbf{O}} \rfloor_i = \boldsymbol{t}', \boldsymbol{t}' \notin \mathsf{W} \;\middle|\; \lfloor \mathbf{A}^{\mathbf{O}} \rfloor_{i-1} = \boldsymbol{t}, \boldsymbol{t} \notin \mathsf{W} \right], \tag{18}$$

where $\boldsymbol{t}' = \boldsymbol{t} \| ((x_i, y_i))$. The resulting labeling P_T maintains the property that the sum of the labels on any non-leaf node and all edges emanating from it equals one.

In Fig. 2b we illustrate a probability labeling associated to $\mathbf{A}_H$ and H. In this case the probability that $\mathbf{A}_H$ wins on its first query is 0.1. The probability that $\mathbf{A}_H$ does not win on its first query and $\mathbf{R}(x_\alpha) = \alpha$, is 0.4, etc.

Given a probability labeling P_T for T, we can assign a value ν to each node in T. If the node v is a leaf node, then its value is the labeling of the node, $P_T(v)$. Otherwise, let $c_1, c_2, \ldots, c_k$ denote v's children, where the label of the edge connecting v to c_i is e_i. Letting $\boldsymbol{c} := (\nu(c_1), \nu(c_2), \ldots, \nu(c_k))$ and $\boldsymbol{e} = (e_1, e_2, \ldots, e_k)$, the value of the node v is defined as

$$\nu(v) := P_T(v) + \boldsymbol{e} \cdot \boldsymbol{c}. \tag{19}$$

The value of a tree T for an adversary $\mathbf{A}$ is defined as the value of the root node. It is easy to see by an inductive argument across the levels of T that the value of T equals the advantage of $\mathbf{A}$. Fig. 2b displays the values of the nodes associated to the labeling of $\mathbf{A}_H$ and H.

Probability Labeling of Multi-Oracle Trees. The nodes in a decision tree T corresponding to a deterministic multi-oracle adversary $\mathbf{A}$ playing game $\overline{G}$ fix the oracles that $\mathbf{A}$ queries at each step. This fact can be used to simplify the labeling P_T for multi-oracle adversaries. Given a node x_i in T, we know that $P_T(x_i)$ equals

$$\mathbb{P}\left[\lfloor \mathbf{A}^{\overline{\mathbf{O}}} \rfloor_i \in \overline{\mathsf{W}} \;\middle|\; \lfloor \mathbf{A}^{\overline{\mathbf{O}}} \rfloor_{i-1} = \boldsymbol{t}, \boldsymbol{t} \notin \overline{\mathsf{W}} \right], \tag{20}$$

where $\boldsymbol{t}$ is the transcript of length $i-1$ associated to x_i. Say that x_i is a query to oracle $\mathbf{O}_j$. Then we can interpret $\mathbf{A}$ interacting with $\overline{\mathbf{O}}$ during this query as being equivalent to a single-query adversary $\mathbf{B}$ interacting with only $\mathbf{O}_j$, such that

$$\mathbb{P}\left[\lfloor \mathbf{A}^{\overline{\mathbf{O}}} \rfloor_i \in \overline{\mathsf{W}} \;\middle|\; \lfloor \mathbf{A}^{\overline{\mathbf{O}}} \rfloor_{i-1} = \boldsymbol{t}, \boldsymbol{t} \notin \overline{\mathsf{W}} \right] = \mathbb{P}\left[\boldsymbol{t} \| \lfloor \mathbf{B}^{\mathbf{O}_j} \rfloor_1 \in \overline{\mathsf{W}} \;\middle|\; \overline{\mathbf{O}}\langle \boldsymbol{t} \rangle, \boldsymbol{t} \notin \overline{\mathsf{W}} \right], \tag{21}$$

where we have replaced the event $\lfloor \mathbf{A}^{\overline{\mathbf{O}}} \rfloor_{i-1} = \boldsymbol{t}$ by $\overline{\mathbf{O}}\langle \boldsymbol{t} \rangle$ since $\mathbf{A}$ is deterministic. Simplifying further, note that $\boldsymbol{t} \| \lfloor \mathbf{B}^{\mathbf{O}_j} \rfloor_1 \in \overline{\mathsf{W}}$ if and only if $\Pi_j \left(\boldsymbol{t} \| \lfloor \mathbf{B}^{\mathbf{O}_j} \rfloor_1 \right) \in \mathsf{W}_j$ conditioned on the fact that $\boldsymbol{t} \notin \overline{\mathsf{W}}$, which means we can focus on

$$\mathbb{P}\left[\Pi_j \left(\boldsymbol{t} \| \lfloor \mathbf{B}^{\mathbf{O}_j} \rfloor_1 \right) \in \mathsf{W}_j \;\middle|\; \overline{\mathbf{O}}\langle \boldsymbol{t} \rangle, \boldsymbol{t} \notin \overline{\mathsf{W}} \right] . \tag{22}$$

The event on the left hand side above is independent of all games except for G_j, and so the above probability equals

$$\mathbb{P}\left[\Pi_j \left(\boldsymbol{t} \| \lfloor \mathbf{B}^{\mathbf{O}_j} \rfloor_1 \right) \in \mathsf{W}_j \;\middle|\; \mathbf{O}_j\langle \Pi_j \boldsymbol{t} \rangle, \Pi_j \boldsymbol{t} \notin \mathsf{W}_j \right] . \tag{23}$$

This means that the label of a node x_i only depends on the particular oracle that to which x_i is queried. We call $\Pi_j \boldsymbol{t}$ the *effective transcript* associated to x_i, since those are the only queries from the transcript which affect $P_T(x_i)$.

From Multi-Oracle to Single-Oracle Trees. Consider Fig. 2c, which depicts the decision tree of an optimal multi-oracle adversary $\mathbf{A}_{\overline{H}}$ playing $\overline{H}$, the multi-oracle version of H. Instead of placing the oracle-input values in each node, we now write the index of the oracle that the adversary queries, so a node containing 2 is a query to $\mathbf{R}_2$. Since all oracles share the same output domain $\{\alpha, \beta\}$, the edges remain the same as in Fig. 2a. In particular, we will continue to name the nodes by their labels in Fig. 2a.

Consider query $x_{\alpha\alpha}$ in Fig. 2c. Since $\mathbf{A}_{\overline{H}}$ has decided to make it a query to $\mathbf{R}_2$, but this is the first query to $\mathbf{R}_2$ on the path containing $x_{\alpha\alpha}$, the effective transcript of that node is empty. In contrast, if $x_{\alpha\alpha}$ would have been an $\mathbf{R}_1$-query, then its effective transcript would have had length two, since $x_{\alpha\alpha}$'s associated transcript contains only $\mathbf{R}_1$-queries. Assuming H is progressive, then there is an optimal adversary $\mathbf{C}_H$ making a single query to $\mathbf{R}_1$ with advantage greater than or equal to the value of node $x_{\alpha\alpha}$. Therefore, we can construct an adversary where $x_{\alpha\alpha}$ is replaced by a query to $\mathbf{R}_1$ without decreasing $\nu(x_{\alpha\alpha})$.

The same reasoning does not hold for the query $x_{\beta\alpha}$, since the effective transcript of that node has length one regardless of whether $\mathbf{R}_1$ or $\mathbf{R}_2$ is queried. However, we do know that if x_β had been an $\mathbf{R}_1$-query, then an optimal choice for $x_{\beta\alpha}$ would have been to query $\mathbf{R}_1$ again since the effective transcript of $\mathbf{R}_1$ would have been longer than the effective transcript of $\mathbf{R}_2$. In particular, consider the decision tree U_H constructed as follows: stick to oracle $\mathbf{R}_1$, and behave as $\mathbf{A}_{\overline{H}}$ does until $\mathbf{A}_{\overline{H}}$ no longer queries $\mathbf{R}_1$, then for each node, compute an optimal choice of oracle input based on the associated transcript up to that point. Assuming H is progressive, we have that

1. $P_{T_H}(x) = P_{U_H}(x)$, $P_{T_H}(x_\alpha) = P_{U_H}(x_\alpha)$, and $P_{T_H}(x_{\alpha\beta}) = P_{U_H}(x_{\alpha\beta})$, since U_H is the same as T_H for those queries, and
2. $P_{T_H}(x_{\alpha\alpha}) \leq P_{U_H}(x_{\alpha\alpha})$ and

$$\begin{aligned} &P_{T_H}(x_\beta) \leq P_{U_H}(x_\beta), P_{T_H}(x_\beta) \leq P_{U_H}(x_\alpha), P_{T_H}(x_{\beta\alpha}) \leq P_{U_H}(x_{\beta\alpha}), \\ &P_{T_H}(x_{\beta\beta}) \leq P_{U_H}(x_{\beta\beta}), P_{T_H}(x_{\beta\beta}) \leq P_{U_H}(x_{\beta\alpha}), \text{ and } P_{T_H}(x_{\beta\alpha}) \leq P_{U_H}(x_{\beta\beta}) , \end{aligned} \tag{24}$$

since the effective transcripts of these nodes in U_H are always longer than their effective transcripts in T_H.

In short, for each subtree S of T_H starting with an $\mathbf{R}_2$-query, the value of each node in a given level ℓ of the corresponding subtree V in U_H is greater than or equal to the value of each node in the same level ℓ of S. Using Lem. 2 below, we can conclude that the value of x_β in T_H is less than or equal to the value of x_β in U_H, and finally that T_H's value is never greater than U_H's value.

The above reasoning can be extended to arbitrary decision trees for a progressive game G. Consider a multi-oracle decision tree T and a single-oracle decision tree U which is the same as T but departs from T the moment T does not make an $\mathbf{O}_1$-query; from that point on U optimizes its next queries only based on effective transcripts. Without loss of generality assume that T contains an $\mathbf{O}_1$-query for its root node. Let S be a subtree of T such that its root node is not an $\mathbf{O}_1$-query, and is the only non-$\mathbf{O}_1$ query on its path connecting it to the root of T. Let V be the corresponding subtree in U. Then, by virtue of G being progressive, given a node s in S and v in V at level ℓ, we know that the effective transcript of s is longer than that of v, and applying Eq. (15), we know that $P_U(s) \geq P_T(v)$. Therefore the probability label of each node in V in that level is greater than or equal to all probability labels in S at the same level. Applying Lem. 2 below, we get our desired result.

To establish Lem. 2, we first need the following result.

Lemma 1. *Let $a \in [0,1]$ and $\boldsymbol{a}^1 \in [0,1]^+$ be such that $a + \mathbf{1} \cdot \boldsymbol{a}^1 = 1$; define b and $\boldsymbol{b}^1$ similarly. Say that $a \geq b$. Let $\boldsymbol{a}^2, \boldsymbol{b}^2 \in [0,1]^+$ with $\min_i a_i^2 \geq \max_i b_i^2$, then*

$$a + \boldsymbol{a}^1 \cdot \boldsymbol{a}^2 \geq b + \boldsymbol{b}^1 \cdot \boldsymbol{b}^2 \,. \tag{25}$$

Proof. Let $a^* = \min_i a_i^2$ and $b^* = \max_i b_i^2$, then

$$\boldsymbol{b}^1 \cdot \boldsymbol{b}^2 \leq b^* \mathbf{1} \cdot \boldsymbol{b}^1 \,, \tag{26}$$

and

$$a^* \mathbf{1} \cdot \boldsymbol{a}^1 \leq \boldsymbol{a}^1 \cdot \boldsymbol{a}^2 \,, \tag{27}$$

therefore

$$\begin{aligned} b + \boldsymbol{b}^1 \cdot \boldsymbol{b}^2 &\leq b + b^* \mathbf{1} \cdot \boldsymbol{b}^1 && (28)\\ &= b + b^*(1-b) && (29)\\ &= b^* + (1-b^*)b && (30)\\ &\leq b^* + (1-b^*)a && (31)\\ &= a + b^* \mathbf{1} \cdot \boldsymbol{a}^1 && (32)\\ &\leq a + a^* \mathbf{1} \cdot \boldsymbol{a}^1 && (33)\\ &\leq a + \boldsymbol{a}^1 \cdot \boldsymbol{a}^2 \,. && (34) \end{aligned}$$

□

Lemma 2. *Let S and V be decision trees with the same number of levels. Let $v_1^\ell, v_2^\ell, \ldots$ and $s_1^\ell, s_2^\ell, \ldots$ denote the nodes of V and S in level ℓ, respectively. Say that for all levels ℓ, we have that $\min_i P_V(v_i^\ell) \geq \max_j P_S(s_i^\ell)$. Then $\nu(V) \geq \nu(S)$.*

Proof. We induct by level of the tree. Our inductive hypothesis is that $\min_i \nu(v_i^\ell) \geq \max_j \nu(s_j^\ell)$, and we want to show that it holds for level $\ell - 1$. However, applying Lemma 1, we get the desired result. □

4 Proving the Absence of Multi-Key Degradation

4.1 Notation and Definitions

We continue to use the notation and definitions from Sect. 3, along with the following.

We use the prefix "multi" to refer to the multi-key setting of the algorithms in question. So, for example, the PRP-PRF switch becomes the multi-PRP-PRF switch, and GCM becomes multi-GCM.

An adversary is *non-adaptive* if the oracle inputs it generates are independent of all oracle outputs. We identify such adversaries with sequences $\boldsymbol{x} \in \mathsf{X}^+$ and write $\mathrm{adv}_G \boldsymbol{x}$ to mean the advantage of the non-adaptive adversary which queries $\boldsymbol{x}$ to win game G.

A *distinguisher* $\mathbf{D}$ is an adversary $\mathbf{A}$ together with a random variable $\mathsf{W} \subset (\mathsf{X} \times \mathsf{Y})^+$, where $\mathbf{A}$ interacts with oracles from X to Y. The advantage of $\mathbf{D}$ in distinguishing oracles $\mathbf{O}_1$ and $\mathbf{O}_2$ is given by

$$\underset{\mathbf{D}}{\Delta}(\mathbf{O}_1 ; \mathbf{O}_2) := \left| \mathbb{P}\left[\mathbf{A}^{\mathbf{O}_1} \in \mathsf{W}\right] - \mathbb{P}\left[\mathbf{A}^{\mathbf{O}_2} \in \mathsf{W}\right] \right| . \tag{35}$$

Note that this definition is equivalent to the usual definition, where the distinguisher's output bit has been changed to the set W, which is some random variable that may depend on $\mathbf{A}$ but is independent of the oracle: $\mathbf{A}^{\boldsymbol{y}} \in \mathsf{W}$ if and only if $\mathbf{A}^{\boldsymbol{y}}$ outputs one, for all possible sequences of oracle outputs $\boldsymbol{y}$.

A uniformly distributed random function (URF) with domain X and range Y is a random variable that is uniformly distributed over the set of all functions from X to Y. A uniformly distributed random permutation (URP) with domain X is a random variable that is uniformly distributed over the set of all permutations on X.

4.2 Non-Adaptivity and the Multi-PRP-PRF Switch

The PRP-PRF switching lemma bounds the distinguishing advantage between a URP π with domain X and a URF ϕ with domain and range X. The lemma states that for all distinguishers $\mathbf{D}$ making no more than q queries,

$$\underset{\mathbf{D}}{\Delta}(\pi ; \phi) \leq \frac{q^2}{2\,|\mathsf{X}|} . \tag{36}$$

Various papers have proofs of this statement, such as [10,19,36]. The corresponding multi-oracle indistinguishability game is

$$\underset{\mathbf{D}}{\Delta}\left(\{\pi_i\}_{i\in I}\ ;\ \{\phi_i\}_{i\in I}\right)\ . \tag{37}$$

Using the generic bound from Sect. 3.3 we get

$$\underset{\mathbf{D}}{\Delta}\left(\{\pi_i\}_{i\in I}\ ;\ \{\phi_i\}_{i\in I}\right) \le |I|\,\frac{q^2}{2\,|\mathsf{X}|}\,, \tag{38}$$

which deteriorates according to the number of oracles present, $|I|$.

Adaptivity does not help adversaries in distinguishing a URP from a URF, as shown for example by Maurer [36]. However, this does not help to prove that there is no degradation in the multi-oracle setting, since non-adaptivity being optimal in the single-oracle setting does not imply that non-adaptivity is still optimal in the multi-oracle setting; Demay et al. [23] construct an example to illustrate this fact, and it can also be seen by considering the weak key example from Sect. 2, where the best strategy in the single-oracle setting is non-adaptive.

Demay et al. [23] also prove that if the oracles in the indistinguishability game satisfy some condition (*conditional equivalence*), which URPs and URFs do, then optimality of non-adaptivity in the multi-oracle setting can be established. However, even if non-adaptive adversaries are optimal in the multi-oracle setting, they can still gain advantage over single-oracle adversaries. Consider for example some game G where adversaries win with probability $1/2^{i+100}$ on the ith query, regardless of what the queries are, and independently of the other queries. In the single-oracle setting adaptivity does not help, and the advantage of any adversary is roughly $2^{-100}(2^q-1)/2^q$. Similarly, in the multi-oracle setting adaptivity does not help, but an adversary with access to μ oracles can achieve an advantage of roughly $\mu 2^{-100}(2^{q/\mu}-1)/2^{q/\mu}$ if they make q/μ queries to each oracle, which approaches $\mu 2^{-100}$ if q/μ is relatively large.

Nevertheless, assuming non-adaptivity in the multi-oracle setting allows us to identify a simpler requirement on games than being progressive. The following result establishes exactly when multi-oracle adversaries have no advantage over single-oracle adversaries when adaptivity does not help.

Proposition 1. *Suppose that $G = (\mathbf{O}, \mathsf{W})$ is a game with optimal non-adaptive adversaries in the multi-oracle setting. Suppose also that for all q and $q' \le q$,*

$$\sup_{\mathbf{A}} \underset{G,q'}{\mathrm{adv}}\,\mathbf{A} + \sup_{\mathbf{A}} \underset{G,q-q'}{\mathrm{adv}}\,\mathbf{A} \le \sup_{\mathbf{A}} \underset{G,q}{\mathrm{adv}}\,\mathbf{A}\,. \tag{39}$$

Then adversaries gain no advantage in interacting with multiple independent instances of G.

Proof. Let $\mathbf{A}$ be a non-adaptive multi-oracle adversary. Let $\mathbf{A}_i := \Pi_i(\mathbf{A})$ and say that $q_i = |\mathbf{A}_i|$. Then we can bound $\mathbf{A}$'s advantage with

$$\underset{\overline{G},Q}{\mathrm{adv}}\,\mathbf{A} \le \sum_{i=1}^{Q} \underset{G,q_i}{\mathrm{adv}}\,\mathbf{A}_i\,. \tag{40}$$

By assumption we know that there is a single-oracle adversary $\mathbf{B}_{1,2}$ making $q_1 + q_2$ queries such that

$$\operatorname*{adv}_{G,q_1} \mathbf{A}_1 + \operatorname*{adv}_{G,q_2} \mathbf{A}_2 \leq \operatorname*{adv}_{G,q_1+q_2} \mathbf{B}_{1,2}\,. \tag{41}$$

The same can be done with $\mathbf{B}_{1,2}$ and $\mathbf{A}_3$ to create adversary $\mathbf{B}_{1,2,3}$, and so on, resulting in a single-oracle adversary which has advantage greater than or equal to $\mathbf{A}$. □

Maurer [36] proved conditional equivalence of URPs and URFs. Therefore adaptivity does not help distinguishers in the single-oracle PRP-PRF switch. Demay et al. [23] proved that conditional equivalence in the single-oracle setting translates to conditional equivalence in the multi-oracle setting. Therefore multi-oracle URPs and URFs are conditionally equivalent, and hence adaptivity does not help in distinguishing multiple URPs from multiple URFs. In particular, distinguishing URPs from URFs is equivalent to finding collisions in URFs [36], which translates indistinguishability into a collision finding game G. Since the advantage in finding a collision in a URF equals the probability that there is some collision among q independent, uniformly distributed elements, the condition in Eq. (39) is satisfied, hence there is no multi-oracle degradation for the PRP-PRF switch.

4.3 Hoang and Tessaro's Technique and an Improvement

Instead of using Prop. 1, one can prove a similar result about the multi-PRP-PRF switch by using the technique of Hoang and Tessaro (HT) [29]. The HT-condition requires understanding the difference in transcript probabilities between a URP and a URF. Let $\boldsymbol{t}$ be a transcript of length q, and say there exists a function $\epsilon(q)$ such that

$$\mathbb{P}\left[\pi\langle\boldsymbol{t}\rangle\right] \geq \mathbb{P}\left[\phi\langle\boldsymbol{t}\rangle\right] \cdot (1 - \epsilon(q))\,. \tag{42}$$

Hoang and Tessaro call this *ϵ-point-wise proximity* of π and ϕ, and we say that ϕ is ϵ-point-wise close to π.. If $\epsilon(q') + \epsilon(q - q') \leq \epsilon(q)$ and $\epsilon(q) \leq 0.5$, then their Lemma 2 establishes that the analogous difference in multi-oracle transcripts is at most $2 \cdot \epsilon(q)$. Following either Hoang and Tessaro's [29] or Chang and Nandi's [19] proof for the PRP-PRF switch, the HT-condition establishes that multi-oracle adversaries have at most a factor of two gain over single-oracle adversaries.

In fact, with only the requirement that $\epsilon(q') + \epsilon(q - q') \leq \epsilon(q)$, one can prove that adversaries gain *no* —not even a factor 2— advantage in the multi-oracle setting relative to ϵ.

Proposition 2. *Suppose that* $\mathbf{R}$ *and* $\mathbf{S}$ *are ϵ-point-wise close and that for all q and $q' \leq q$, $\epsilon(q') + \epsilon(q - q') \leq \epsilon(q)$. Then* $\overline{\mathbf{R}}$ *and* $\overline{\mathbf{S}}$*, which are oracles giving adversaries access to arbitrarily many independent instances of* $\mathbf{R}$ *and* $\mathbf{S}$*, are ϵ-point-wise close as well.*

Proof. It suffices to prove that for all $\boldsymbol{t}$ such that $|\boldsymbol{t}| = q$ and $\mathbb{P}\left[\overline{\mathbf{S}}\langle \boldsymbol{t} \rangle\right] > 0$,

$$\frac{\mathbb{P}\left[\overline{\mathbf{R}}\langle \boldsymbol{t} \rangle\right]}{\mathbb{P}\left[\overline{\mathbf{S}}\langle \boldsymbol{t} \rangle\right]} \geq 1 - \epsilon(q)\,. \tag{43}$$

Let I be the set of instances queried in $\boldsymbol{t}$, and say that $q_i = |\Pi_i \boldsymbol{t}|$, then

$$\frac{\mathbb{P}\left[\overline{\mathbf{R}}\langle \boldsymbol{t} \rangle\right]}{\mathbb{P}\left[\overline{\mathbf{S}}\langle \boldsymbol{t} \rangle\right]} = \prod_{i \in I} \frac{\mathbb{P}\left[\mathbf{R}_i\langle \Pi_i \boldsymbol{t} \rangle\right]}{\mathbb{P}\left[\mathbf{S}_i\langle \Pi_i \boldsymbol{t} \rangle\right]} \geq \prod_{i \in I}(1 - \epsilon(q_i)) \geq 1 - \sum_{i \in I} \epsilon(q_i) \geq 1 - \epsilon(q)\,. \tag{44}$$

□

An important difference between our setting and Hoang and Tessaro's is that our oracles are independent of each other, whereas Hoang and Tessaro also consider oracles which are built using some shared underlying ideal primitive, which is why Prop. 2 cannot be applied to their setting.

The condition that $\epsilon(q') + \epsilon(q - q') \leq \epsilon(q)$ looks very similar to the condition of Prop. 1 required in order to achieve no multi-oracle degradation when adaptivity does not help, since ϵ is an upper bound on the success probability of single-oracle adversaries. However, Prop. 2 is a statement about the computed bounds, and it might be the case that multi-oracle adversaries have some advantage gain over single-oracle adversaries, but that this difference is not visible with a particular upper bound ϵ; after all, setting $\epsilon(q) = q$ is true for all pairs of oracles, but then Prop. 2 becomes meaningless. In contrast, satisfying the hypotheses of Prop. 1, and, more generally, a game being progressive, establishes something inherent about the oracle in question sufficient to prove that multi-oracle adversaries gain nothing over single-oracle adversaries.

4.4 Integrity and the Inapplicability of Point-wise Proximity

Finding meaningful ϵ which establishes point-wise-proximity is impossible in some cases, as we illustrate for MAC (Message Authentication Code) schemes and integrity. Our focus is on stateful MAC schemes, although the same observations can be applied to deterministic MAC schemes.

Definition 4. *A nonce-based MAC scheme from* $\mathsf{N} \times \mathsf{M}$ *to* T *is a pair of algorithms* (F, V)*, where* F*, the tagging algorithm, maps a tuple of a nonce from* N *and message from* M *to tags in* T*, and* V*, the verification algorithm, maps inputs from* $\mathsf{N} \times \mathsf{M} \times \mathsf{T}$ *to either* $\top$ *or* $\bot$*, indicating validity or invalidity of an input.*

A secure MAC scheme is one in which it is difficult to construct a new input to V such that V outputs $\top$. We translate Bellare and Namprempre's strong unforgeability [8] into our formalization.

Definition 5. *Let (F, V) be a nonce-based MAC scheme. The integrity game G with respect to (F, V) is defined as $(\mathbf{O}, \mathsf{W})$, with $\mathbf{O}$ an oracle giving adversaries access to F and V, and W defined as the set of transcripts consisting of F-queries where each nonce-input is unique, and containing at least one V-query (n, m, t) where $V(n, m, t) = \top$, and $F(n, m) = t$ is not in the preceding transcript.*

Recall that adversarial advantage is defined as in Eq. (4).

In order to use pointwise proximity in an integrity game G, it needs to be written as an indistinguishability game, which is done as follows:

$$\Delta\left(F, V\,;\, F, \bot\right), \tag{45}$$

with $\bot$ an algorithm always outputting $\bot$. Establishing ϵ-point-wise proximity between (F, V) and $(F, \bot)$ means finding an ϵ such that for all transcripts $\boldsymbol{t}$ of length q,

$$\mathbb{P}\left[(F, V)\langle \boldsymbol{t} \rangle\right] \geq (1 - \epsilon) \cdot \mathbb{P}\left[(F, \bot)\langle \boldsymbol{t} \rangle\right], \tag{46}$$

where we write (F, V) and $(F, \bot)$ as shorthands for oracles. Letting O denote either V or $\bot$, the transcript consisting of $O(n, m, t) = \bot$ followed by $F(n, m) = t$ has zero probability with (F, V) and non-zero probability with $(F, \bot)$, meaning ϵ must equal one. Swapping (F, V) and $(F, \bot)$ in Eq. (46) causes the same problem with any transcript containing an $O(n, m, t) = \top$ query. Therefore, ϵ-point-wise proximity can only hold for $\epsilon = 1$, making the bounds obtained with ϵ-point-wise proximity vacuous.

4.5 Bernstein's Theorem in the Multi-Oracle Setting

Rather than considering indistinguishability, ϵ-pointwise proximity can be directly applied to games themselves, as is done by Bernstein [12, 13], where ϵ-pointwise proximity is called interpolation probability. Bernstein shows that the probability that an adversary outputs 1 when interacting with an oracle which is ϵ-pointwise close to a URF, is at most $(1 - \epsilon)^{-1}$ times the probability the adversary outputs one when interacting with a URF. Bernstein replaces the use of the PRP-PRF switch with his result when computing integrity bounds for MACs, thereby significantly improving them. Iwata, Ohashi, and Minematsu apply this technique to GCM as well [33, Section 7.5 and Appendix C].

Although Bernstein only considers the special case in which one of the oracles is a URF, it can be easily generalized to any oracle. We state the result in terms of distinguishers, which is equivalent to considering adversaries with binary output. Note that this means the result is only applicable to games where W is independent of the oracle $\mathbf{O}$.

Theorem 2. *Let $\mathbf{D} = (\mathbf{A}, \mathsf{W})$ be any distinguisher and q a positive integer, then if $\mathbf{O}_1$ is ϵ-pointwise close to $\mathbf{O}_2$,*

$$\mathbb{P}\left[\mathbf{A}^{\mathbf{O}_1} \in \lfloor \mathsf{W} \rfloor_q\right] \leq (1 - \epsilon(q))^{-1} \cdot \mathbb{P}\left[\mathbf{A}^{\mathbf{O}_2} \in \lfloor \mathsf{W} \rfloor_q\right]. \tag{47}$$

Proof. Without loss of generality, assume that $\mathbf{A}$ makes exactly q queries, as one can always consider a distinguisher $\mathbf{D}'$ instead which runs $\mathbf{A}$, makes exactly q queries, and ignores the additional query-outputs.

$$\mathbb{P}\left[\mathbf{A}^{\mathbf{O}_1} \in \lfloor \mathsf{W} \rfloor_q\right] = \sum_{|\boldsymbol{t}|=q} \mathbb{P}\left[\mathbf{A}\langle \boldsymbol{t}\rangle, \boldsymbol{t} \in \mathsf{W}\right] \cdot \mathbb{P}\left[\mathbf{O}_1\langle \boldsymbol{t}\rangle\right] \tag{48}$$

$$\leq (1-\epsilon(q))^{-1} \cdot \sum_{|\boldsymbol{t}|=q} \mathbb{P}\left[\mathbf{A}\langle \boldsymbol{t}\rangle, \boldsymbol{t} \in \mathsf{W}\right] \cdot \mathbb{P}\left[\mathbf{O}_2\langle \boldsymbol{t}\rangle\right] \tag{49}$$

$$= (1-\epsilon(q))^{-1} \cdot \mathbb{P}\left[\mathbf{A}^{\mathbf{O}_2} \in \lfloor \mathsf{W} \rfloor_q\right] . \tag{50}$$

□

Bernstein's theorem can be applied to the multi-oracle setting using Prop. 2: if $\mathbf{O}_1$ is ϵ-pointwise close to $\mathbf{O}_2$, and ϵ satisfies the hypothesis of Prop. 2, then the above result can be applied to $\overline{\mathbf{O}}_1$ and $\overline{\mathbf{O}}_2$. For example, this holds in the case of URPs and URFs, hence Bernstein's theorem can be applied to multi-URPs and multi-URFs as well.

Corollary 1. *Let $\mathbf{D} = (\mathbf{A}, \mathsf{W})$ be any distinguisher and q a positive integer. Let π denote a URP and ϕ a URF, with $\overline{\pi}$ and $\overline{\phi}$ their multi-oracle counterparts, then*

$$\mathbb{P}\left[\mathbf{A}^{\overline{\pi}} \in \lfloor \mathsf{W} \rfloor_q\right] \leq (1-\epsilon(q))^{-1} \cdot \mathbb{P}\left[\mathbf{A}^{\overline{\phi}} \in \lfloor \mathsf{W} \rfloor_q\right] , \tag{51}$$

where ϵ is the proximity function of π and ϕ.

4.6 Multi-Wegman-Carter Security

Wegman-Carter authenticators [52] are nonce-based MAC schemes mapping messages in M to tags in T. The tagging algorithm takes a nonce $n \in \mathsf{N}$ and a message $m \in \mathsf{M}$, and maps (n, m) to $\phi(n) + h(m)$, where T is a group, ϕ is a URF, and $h : \mathsf{M} \to \mathsf{T}$ is a random function for which it is difficult to find collisions. The verification algorithm takes a nonce $n \in \mathsf{N}$, a message $m \in \mathsf{M}$, and a tag $t \in \mathsf{T}$, and checks whether (n, m) maps to t; it outputs $\top$ if this is the case, and $\bot$ otherwise.

Usually the security of Wegman-Carter authenticators is proved [34, 46, 52] relative to

$$\sup_{m_1 \neq m_2, t} \mathbb{P}\left[h(m_1) - h(m_2) = t\right] , \tag{52}$$

however we will need to describe h's collision resistance differently in order to characterize when Wegman-Carter authenticators exhibit no multi-oracle degradation.

Definition 6. *Let $h : \mathsf{M} \to \mathsf{T}$ be a random function with T a group. Define the collision game $G = (\mathbf{O}, \mathsf{W})$ where $\mathbf{O} : \mathsf{M}^2 \times \mathsf{T} \to \{\top, \bot\}$ outputs $\top$ on input (m_1, m_2, t) if $h(m_1) - h(m_2) = t$, and $\bot$ otherwise, and W consists of all transcripts containing an $\mathbf{O}$-query (m_1, m_2, t) with $m_1 \neq m_2$ and $\mathbf{O}(m_1, m_2, t) = \top$.*

Proposition 3. *Consider adversaries which make no more than $|\mathsf{N}|$ queries. Then Wegman-Carter authenticators exhibit no multi-oracle degradation with respect to the integrity game from Def. 5 if the underlying random function h exhibits no multi-oracle degradation with respect to the collision game in Def. 6.*

Proof. Let (F, V) denote the Wegman-Carter authenticator and let G be its associated integrity game. Let $\mathbf{A}$ be a multi-oracle adversary playing $\overline{G}$.

First we establish that adversaries gain no advantage by choosing their nonces adaptively. Let $\boldsymbol{n}^i = (n_1^i, n_2^i, \ldots)$ be an enumeration of N, one for each possible oracle $i \in \mathbb{N}$. Then we construct adversary $\mathbf{A}_{\boldsymbol{n}}$ from $\mathbf{A}$ as follows. $\mathbf{A}_{\boldsymbol{n}}$ runs $\mathbf{A}$, and maintains a mapping $\iota : \mathbb{N} \times \mathsf{N} \to \mathbb{N}$ which keeps track of the order in which a particular nonce $n \in \mathsf{N}$ was queried for oracle $i \in \mathbb{N}$; for example if $(3, X)$ is the fifth nonce queried to the third oracle, then $\iota(3, X) = 5$. Each time $\mathbf{A}$ makes an F-query (n, m) to oracle i, $\mathbf{A}_{\boldsymbol{n}}$ makes the F-query $(n^i_{\iota(i,n)}, m)$ to oracle i and returns the response to $\mathbf{A}$. Similarly, each time $\mathbf{A}$ makes a V-query (n, m, t) to oracle i, $\mathbf{A}_{\boldsymbol{n}}$ makes the V-query $(n^i_{\iota(i,n)}, m, t)$ to oracle i and returns the response to $\mathbf{A}$. Since the mapping $n \mapsto n^i_{\iota(i,n)}$ is bijective for each i, $\mathbf{A}_{\boldsymbol{n}}$'s advantage is at least that of $\mathbf{A}$ since the URF ϕ underlying oracle i is indistinguishable from the URF $n \mapsto \phi(n^i_{\iota(i,n)})$. Therefore we restrict our attention to adversaries which choose their nonces non-adaptively.

Consider an adversary interacting in the multi-oracle integrity game. Since adaptivity does not help when picking nonces, and the total number of queries is not greater than $|\mathsf{N}|$, we can force the adversary to pick distinct nonces to query. This allows us to replace all the URFs from each Wegman-Carter authenticator by a single URF, since the inputs to the URF will always be distinct. Therefore, we may restrict our attention to adversaries interacting with multiple Wegman-Carter authenticators using the same URF.

For each nonce n, we let m_n denote the associated message input, and t_n F's output under n, so that $F(n, m_n) = t_n$. To each nonce n we can associate two sets of pairs $R_n, S_n \subset \mathsf{M} \times \mathsf{T}$ where $(m, t) \in R_n$ if there is a V-query $V(n, m, t)$ before the F-query using n as nonce is made, and S_n is all pairs (m, t) from V-queries after the F-query using n as nonce is made; S_n is empty if there is no such F-query. Without loss of generality we can assume that for all queried nonces, $R_n \cup S_n \neq \emptyset$, since otherwise $F(n, m_n)$ is independent of the adversary winning.

A nonce wins if one of its associated verification queries results in $\top$, meaning there exists $(m, t) \in R_n \cup S_n$ such that $V(n, m, t) = \top$. Note that $\phi(n) = t - h(m)$ for $(m, t) \in S_n$ if and only if $h(m_n) - h(m) = t_n - t$, and similarly $h(m_n) - h(m) = t_n - t$ for $(m, t) \in R_n$ if and only if $\phi(n) = t - h(m)$. Therefore, a nonce n wins only if

$$\phi(n) \in \{t - h(m) \mid (m, t) \in R_n\} \tag{53}$$

or there exists $(m, t) \in S_n$ such that

$$h(m_n) - h(m) = t_n - t\,. \tag{54}$$

We call a verification query $V(n, m, t)$ a guess if it occurs before the corresponding F-query with nonce n, and a collision attempt if it occurs after the

F-query. A guess succeeds only if Eq. (53) is satisfied, and a collision attempt succeeds only if Eq. (54) is satisfied.

Let $\mathbf{A}$ be an adversary interacting with multiple Wegman-Carter authenticators using the same URF (always querying distinct nonces to the authenticators), and different random functions h_i for $i \in \mathbb{N}$. The adversary $\mathbf{A}$ either wins with a guess, or a collision attempt.

Say that it is given that $\mathbf{A}$ does not win with a guess. This means that for all n,

$$\phi(n) \notin \{t - h_i(m) \mid (m, t) \in R_n\} \ , \tag{55}$$

and $\mathbf{A}$ wins only if there is a nonce n for which Eq. (54) is satisfied, meaning $\mathbf{A}$ has found a collision for h. We construct an adversary $\mathbf{B}$ playing the multi-oracle collision game with h_i. The adversary $\mathbf{B}$ runs $\mathbf{A}$, responds to $\mathbf{A}$'s guesses with $\perp$, it responds to $\mathbf{A}$'s F-queries by uniformly sampling an element from

$$\mathsf{T} \setminus \{t \mid (m, t) \in R_n\} \ , \tag{56}$$

and $\mathbf{B}$ responds to $\mathbf{A}$'s collision attempts $V(n, m, t)$ by querying $(m_n, m, t_n - t)$ to the appropriate oracle ($\mathbf{O}_i$ if h_i was queried), where $F(n, m_n) = t_n$. Then, given that $\mathbf{A}$ does not win with a guess, $\mathbf{B}$ perfectly simulates $\mathbf{A}$'s game since all of $\mathbf{A}$'s guesses fail, F is distributed correctly given that all of $\mathbf{A}$'s guesses fail, and $\mathbf{A}$'s collision attempts are passed directly to the collision oracles.

By hypothesis, we know that for every $i > 0$ there is an adversary $\mathbf{C}_i$ playing the collision game with one random function h such that $\mathrm{adv}_i\, \mathbf{C}_i \geq \mathrm{adv}_i\, \mathbf{B}$, and in particular $\mathrm{adv}_i\, \mathbf{C}_i$ is greater than or equal to the probability that $\mathbf{B}$ wins and makes i queries.

Using $\mathbf{C}_i$ and $\mathbf{A}$, we construct a single-oracle adversary $\mathbf{A}_1$ playing the Wegman-Carter integrity game. First $\mathbf{A}_1$ runs $\mathbf{A}$ and responds to $\mathbf{A}$'s queries using its own independently simulated Wegman-Carter authenticators. Once $\mathbf{A}$ is finished, $\mathbf{A}_1$ takes all of $\mathbf{A}$'s guesses and forwards them to its own oracle. Then, if $\mathbf{A}_1$ does not win with a guess, it computes how many queries i it has remaining, and then runs $\mathbf{C}_i$. The probability that $\mathbf{A}$ wins with a guess equals the probability that $\mathbf{A}_1$ wins with a guess, since it is the probability that ϕ gets mapped into the sets defined in Eq. (53).

The probability that $\mathbf{A}$ makes i non-guess queries and wins, given that $\mathbf{A}$'s guesses fail, is bounded by the probability that $\mathbf{B}$ wins and makes i queries, which in turn is bounded by $\mathrm{adv}_i\, \mathbf{C}_i$. Therefore the probability that $\mathbf{B}$ wins is bounded by the sum of $p_i \cdot \mathrm{adv}_i\, \mathbf{C}_i$, where p_i is the probability that $\mathbf{A}_1$ has i queries remaining after its guesses. Since the sum of the p_i is 1, we know that the probability that $\mathbf{A}_1$ wins given that its guesses fail is greater than or equal to the probability that $\mathbf{B}$ wins. Therefore we have shown that the single-oracle adversary $\mathbf{A}_1$ has no less advantage than the multi-oracle adversary $\mathbf{A}$. □

4.7 Multi-GCM Security

Given the results in the previous sections, it is straightforward to prove that GCM does not have bounds which increase as μ increases. We give a brief description of GCM with 96 bit nonces, which is the one used by TLS; a complete

description of GCM can be found in the original document [37] or the analysis by Iwata, Ohashi, and Minematsu [32]. We also refer to Iwata et al. for the definitions of confidentiality and integrity for GCM.

GCM uses a block cipher $E : \mathsf{K} \times \mathsf{X} \to \mathsf{X}$, where $\mathsf{X} = \{0,1\}^{128}$, however, using standard arguments, we can focus on GCM using a URP π over X instead. GCM[π] consists of an encryption enc and a decryption algorithm dec where

$$\mathsf{enc} : \mathsf{N} \times \mathsf{H} \times \mathsf{M} \to \mathsf{C}\,, \tag{57}$$

$$\mathsf{dec} : \mathsf{N} \times \mathsf{H} \times \mathsf{C} \to \mathsf{M} \cup \{\perp\}\,, \tag{58}$$

with N the nonce space, H the associated data, M the plaintexts, C the ciphertexts, and $\perp$ an error symbol.

On input of (n, a, m), enc generates unique inputs to π, $n^0, n^1, \ldots, n^\ell$. The values n^i for $i > 0$ are used to run CTR mode [41] in order to encrypt the plaintext m. The resulting ciphertext c is then used together with the associated data a, and run through a polynomial hash function $h : \mathbf{A} \times \mathbf{C} \to \mathsf{X}$, also called GHASH. GHASH's output is then XORed together with the output of π under n^0 to create a Wegman-Carter-style authenticator. The polynomial hash h uses $L := \pi(0^{128})$ as a key. GCM with 96 bit nonces ensures that every time π is called by the encryption oracle, π receives a different input.

By applying a PRP-PRF switch to GCM, π is replaced with a URF ϕ, and so the confidentiality of GCM can be bounded by the PRP-PRF switch, as illustrated by Iwata et al. [32]. In the multi-oracle setting a multi-PRP-PRF switch can be performed, thereby establishing that multi-GCM's confidentiality is bounded above by the multi-PRP-PRF switch. As shown previously, the multi-PRP-PRF switch is independent of the number of keys, hence multi-GCM's confidentiality bound is independent of the number of keys.

Rather than applying a PRP-PRF switch for integrity, we can apply Bernstein's theorem, as Iwata, Ohashi, and Minematsu did [33, Section 7.5 and Appendix C]. As a result, one can show that GCM's integrity can be bounded by the integrity of the Wegman-Carter authenticator using GHASH. This is because π is replaced by a URF ϕ, and the inputs to ϕ used in the underlying CTR mode are always distinct from the inputs to ϕ used in the underlying Wegman-Carter authenticator. Therefore, the underlying Wegman-Carter authenticator becomes independent of the underlying CTR mode, and GCM with ϕ is just an Encrypt-then-MAC [8,40] style authenticated encryption algorithm, meaning its integrity bound is bounded above by the integrity of the underlying MAC.

In the same way, by applying Cor. 1, the integrity of multi-GCM can bounded by that of a multi-Wegman-Carter authenticator. Therefore, establishing that GCM's integrity bound does not degrade in the multi-oracle setting can be done by proving that GHASH with respect to the collision game of Def. 6 is progressive. In the lemma below we do exactly this, although we drop out the padding and input formatting from GHASH since it does not significantly affect the analysis below.

Lemma 3. *Let* X *be a finite field and let* $h : \mathsf{X} \times \mathsf{X}^{\leq \ell} \to \mathsf{X}$ *be the function defined by*

$$h(k, \boldsymbol{x}) = \sum_{i=1}^{q} k^i x_{\ell - i}\,, \tag{59}$$

where $|\boldsymbol{x}| = q \leq \ell$*, then if* k *is a uniformly distributed random key over* X*,* $h(k, \cdot)$ *with respect to the collision game* G *of Def. 6 is progressive.*

Proof. Let $\mathbf{A}$ be an adversary playing G, and say that it makes queries (m_1, m_1', t_1), $\ldots$, (m_q, m_q', t_q), then $\mathbf{A}$'s advantage is given by the probability that for some i,

$$h(k, m_i) - h(k, m_i') = t_i\,. \tag{60}$$

The value $h(k, m_i) - h(k, m_i')$ is a polynomial in k of degree $\max\{|m_i|, |m_i'|\}$, hence Eq. (60) defines a set of keys K_i for which the equation holds. In particular, Eq. (60) holds if and only if $k \in K_i$, therefore $\mathbf{A}$'s advantage is the probability that $k \in K_1 \cup \cdots \cup K_q$. A non-winning transcript is a set of inputs for which Eq. (60) does not hold, therefore conditioning on a non-winning transcript of length q' is the same as saying that $k \notin K_1' \cup \cdots \cup K_{q'}'$.

In particular, we can remove some adaptivity from optimal single-oracle adversaries as follows. For each query (m, m', t) which does not result in a collision, the adversary eliminates a set of potential keys, and increases the set B of non-keys, that is, $k \notin B$. Therefore the optimal adversary selects (m, m', t) such that ℓ keys are eliminated for each query (where ℓ is the maximum degree possible of the polynomial). In order to do so, the adversary can just pick elements $r_1, r_2, \ldots, r_{q\ell}$ outside of $\mathsf{X} \setminus B$, reconstruct the polynomials $(\mathbf{k} - r_{(i-1)\ell+1})(\mathbf{k} - r_{(i-1)\ell+2}) \cdots (\mathbf{k} - r_{i\ell})$ for $i = 1, \ldots, q$, where $\mathbf{k}$ is a formal symbol, and from these polynomials reconstruct the corresponding h-queries. In particular, any transcript of length i which is meaningful will eliminate exactly $i \cdot \ell$ keys.

Furthermore, the game is progressive because the longer the transcript given to an optimal adversary, the larger the set of keys which are eliminated, and the greater the chance that a collision occurs. □

This allows us to conclude that GCM's integrity bound does not exhibit multi-oracle degradation, and as a result, we have the following proposition.

Proposition 4. *The confidentiality and integrity bounds for GCM with 96 bit nonces in the multi-key setting are the same as those in the single-key setting as established by Iwata et al. in [32, Corollary 3] and [32, Section 7.5 and Appendix C], respectively.*

Note that there are papers attacking polynomial-based Wegman-Carter authenticators [28, 48], where the attacks focus on finding weak keys. However, as shown by the analysis of Procter and Cid [43, 44], almost every subset of the keyspace can be considered a weak key class. Hence our results do not contradict prior work.

5 Future Work

Although we established that GCM does not exhibit multi-key degradation, there are still many other widely deployed algorithms for which there are as yet no results. Our approach has been to extract an abstract condition which could be applied to any algorithm and which is sufficient for proving the absence of multi-key security degradation. However the condition seems to be quite strong, and there might be other conditions which exactly capture when an algorithm does not suffer from multi-key degradation, possibly only applying to restricted classes of schemes. For example, our condition makes no restriction on whether the algorithm is stateful or stateless, while a condition for stateless algorithms might be simpler, or more powerful. How useful such conditions are remains to be seen, but they would at least fundamentally advance our understanding of the analysis of algorithms, and at best allow us to categorize algorithms according to their multi-key degradation.

References

1. Bader, C., Jager, T., Li, Y., Schäge, S.: On the impossibility of tight cryptographic reductions. In: Fischlin, M., Coron, J. (eds.) EUROCRYPT 2016, Vienna, Austria, May 8-12, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9666, pp. 273–304. Springer (2016)
2. Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., Regazzoni, F.: Midori: A Block Cipher for Low Energy. In: Iwata and Cheon [31], pp. 411–436
3. Bellare, M., Bernstein, D.J., Tessaro, S.: Hash-Function Based PRFs: AMAC and Its Multi-User Security. In: Fischlin, M., Coron, J. (eds.) EUROCRYPT 2016, Vienna, Austria, May 8-12, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9665, pp. 566–595. Springer (2016)
4. Bellare, M., Boldyreva, A., Micali, S.: Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements. Full version. (2000), `http://cseweb.ucsd.edu/~mihir/papers/musu.html`
5. Bellare, M., Boldyreva, A., Micali, S.: Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements. In: Preneel, B. (ed.) EUROCRYPT 2000, Bruges, Belgium, May 14-18, 2000, Proceeding. Lecture Notes in Computer Science, vol. 1807, pp. 259–274. Springer (2000)
6. Bellare, M., Canetti, R., Krawczyk, H.: Pseudorandom Functions Revisited: The Cascade Construction and Its Concrete Security. In: 37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996. pp. 514–523. IEEE Computer Society (1996)
7. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A Concrete Security Treatment of Symmetric Encryption. In: 38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997. pp. 394–403. IEEE Computer Society (1997)
8. Bellare, M., Namprempre, C.: Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In: Okamoto, T. (ed.) ASIACRYPT 2000, Kyoto, Japan, December 3-7, 2000, Proceedings. Lecture Notes in Computer Science, vol. 1976, pp. 531–545. Springer (2000)

9. Bellare, M., Rogaway, P.: Entity Authentication and Key Distribution. In: Stinson, D.R. (ed.) CRYPTO '93, Santa Barbara, California, USA, August 22-26, 1993, Proceedings. Lecture Notes in Computer Science, vol. 773, pp. 232–249. Springer (1993)
10. Bellare, M., Rogaway, P.: Code-Based Game-Playing Proofs and the Security of Triple Encryption. Cryptology ePrint Archive, Report 2004/331 (2004)
11. Bellare, M., Tackmann, B.: The Multi-user Security of Authenticated Encryption: AES-GCM in TLS 1.3. In: Robshaw and Katz [45], pp. 247–276
12. Bernstein, D.J.: Stronger security bounds for permutations (2005), `http://cr.yp.to/papers.html#permutations`, Date accessed 9 April, 2015
13. Bernstein, D.J.: Stronger security bounds for Wegman-Carter-Shoup authenticators. In: Cramer, R. (ed.) EUROCRYPT 2005, Aarhus, Denmark, May 22-26, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3494, pp. 164–180. Springer (2005)
14. Biham, E.: How to decrypt or even substitute DES-encrypted messages in 2^{28} steps. Inf. Process. Lett. 84(3), 117–124 (2002)
15. Biryukov, A., Mukhopadhyay, S., Sarkar, P.: Improved Time-Memory Trade-Offs with Multiple Data. In: Preneel, B., Tavares, S.E. (eds.) Selected Areas in Cryptography, 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers. Lecture Notes in Computer Science, vol. 3897, pp. 110–127. Springer (2005)
16. Blake-Wilson, S., Johnson, D., Menezes, A.: Key Agreement Protocols and Their Security Analysis. In: Darnell, M. (ed.) Cryptography and Coding, 6th IMA International Conference, Cirencester, UK, December 17-19, 1997, Proceedings. Lecture Notes in Computer Science, vol. 1355, pp. 30–45. Springer (1997)
17. Bogdanov, A., Chang, D., Ghosh, M., Sanadhya, S.K.: Bicliques with Minimal Data and Time Complexity for AES. In: Lee, J., Kim, J. (eds.) Information Security and Cryptology - ICISC 2014 - 17th International Conference, Seoul, Korea, December 3-5, 2014, Revised Selected Papers. Lecture Notes in Computer Science, vol. 8949, pp. 160–174. Springer (2014)
18. Bogdanov, A., Khovratovich, D., Rechberger, C.: Biclique Cryptanalysis of the Full AES. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011, Seoul, South Korea, December 4-8, 2011. Proceedings. Lecture Notes in Computer Science, vol. 7073, pp. 344–371. Springer (2011)
19. Chang, D., Nandi, M.: A Short Proof of the PRP/PRF Switching Lemma. Cryptology ePrint Archive, Report 2008/078 (2008), `http://eprint.iacr.org/2008/078`
20. Chatterjee, S., Koblitz, N., Menezes, A., Sarkar, P.: Another Look at Tightness II: Practical Issues in Cryptography. Cryptology ePrint Archive, Report 2016/360 (2016)
21. Chatterjee, S., Menezes, A., Sarkar, P.: Another Look at Tightness. In: Miri, A., Vaudenay, S. (eds.) Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7118, pp. 293–319. Springer (2011)
22. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography, Springer (2002)
23. Demay, G., Gaži, P., Maurer, U., Tackmann, B.: Optimality of non-adaptive strategies: The case of parallel games. In: 2014 IEEE International Symposium on Information Theory, Honolulu, HI, USA, June 29 - July 4, 2014. pp. 1707–1711. IEEE (2014)

24. Fouque, P., Joux, A., Mavromati, C.: Multi-user Collisions: Applications to Discrete Logarithm, Even-Mansour and PRINCE. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I. Lecture Notes in Computer Science, vol. 8873, pp. 420–438. Springer (2014)
25. Gazi, P., Maurer, U.: Free-start distinguishing: Combining two types of indistinguishability amplification. In: Kurosawa, K. (ed.) Information Theoretic Security, 4th International Conference, ICITS 2009, Shizuoka, Japan, December 3-6, 2009. Revised Selected Papers. Lecture Notes in Computer Science, vol. 5973, pp. 28–44. Springer (2009)
26. Guo, J., Jean, J., Nikolić, I., Qiao, K., Sasaki, Y., Sim, S.M.: Invariant Subspace Attack Against Midori64 and The Resistance Criteria for S-box Designs. IACR Transactions on Symmetric Cryptology 1(1), 19 (2017)
27. Guo, Z., Wu, W., Liu, R., Zhang, L.: Multi-key Analysis of Tweakable Even-Mansour with Applications to Minalpher and OPP. IACR Transactions on Symmetric Cryptology 1, 19 (2017), http://eprint.iacr.org/2016/1098
28. Handschuh, H., Preneel, B.: Key-recovery attacks on universal hash function based MAC algorithms. In: Wagner, D. (ed.) CRYPTO 2008, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5157, pp. 144–161. Springer (2008)
29. Hoang, V.T., Tessaro, S.: Key-Alternating Ciphers and Key-Length Extension: Exact Bounds and Multi-user Security. In: Robshaw and Katz [45], pp. 3–32
30. Hoang, V.T., Tessaro, S.: The multi-user security of double encryption. In: Coron, J., Nielsen, J.B. (eds.) EUROCRYPT 2017, Paris, France, April 30 - May 4, 2017, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10211, pp. 381–411 (2017)
31. Iwata, T., Cheon, J.H. (eds.): ASIACRYPT 2015, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II, Lecture Notes in Computer Science, vol. 9453. Springer (2015)
32. Iwata, T., Ohashi, K., Minematsu, K.: Breaking and Repairing GCM Security Proofs. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7417, pp. 31–49. Springer (2012)
33. Iwata, T., Ohashi, K., Minematsu, K.: Breaking and repairing GCM security proofs. IACR Cryptology ePrint Archive 2012, 438 (2012)
34. Krawczyk, H.: LFSR-based Hashing and Authentication. In: Desmedt, Y. (ed.) CRYPTO '94, Santa Barbara, California, USA, August 21-25, 1994, Proceedings. Lecture Notes in Computer Science, vol. 839, pp. 129–139. Springer (1994)
35. Maurer, U.: Indistinguishability of Random Systems. In: Knudsen, L.R. (ed.) EUROCRYPT 2002, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2332, pp. 110–132. Springer (2002)
36. Maurer, U.: Conditional equivalence of random systems and indistinguishability proofs. In: Proceedings of the 2013 IEEE International Symposium on Information Theory, Istanbul, Turkey, July 7-12, 2013. pp. 3150–3154. IEEE (2013)
37. McGrew, D.A., Viega, J.: The Security and Performance of the Galois/Counter Mode (GCM) of Operation. In: Canteaut, A., Viswanathan, K. (eds.) Progress in Cryptology - INDOCRYPT 2004, Chennai, India, December 20-22, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3348, pp. 343–355. Springer (2004)
38. Menezes, A., Smart, N.P.: Security of Signature Schemes in a Multi-User Setting. Des. Codes Cryptography 33(3), 261–274 (2004)

39. Mouha, N., Luykx, A.: Multi-key Security: The Even-Mansour Construction Revisited. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9215, pp. 209–223. Springer (2015)
40. Namprempre, C., Rogaway, P., Shrimpton, T.: Reconsidering generic composition. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014, Copenhagen, Denmark, May 11-15, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8441, pp. 257–274. Springer (2014)
41. National Institute of Standards and Technology: DES Modes of Operation. FIPS 81 (December 1980)
42. Niwa, Y., Ohashi, K., Minematsu, K., Iwata, T.: GCM Security Bounds Reconsidered. In: Leander, G. (ed.) FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers. Lecture Notes in Computer Science, vol. 9054, pp. 385–407. Springer (2015)
43. Procter, G., Cid, C.: On weak keys and forgery attacks against polynomial-based MAC schemes. In: Moriai, S. (ed.) FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers. Lecture Notes in Computer Science, vol. 8424, pp. 287–304. Springer (2013)
44. Procter, G., Cid, C.: On weak keys and forgery attacks against polynomial-based MAC schemes. J. Cryptology 28(4), 769–795 (2015)
45. Robshaw, M., Katz, J. (eds.): CRYPTO 2016, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I, Lecture Notes in Computer Science, vol. 9814. Springer (2016)
46. Rogaway, P.: Bucket Hashing and Its Application to Fast Message Authentication. J. Cryptology 12(2), 91–115 (1999)
47. Rogaway, P.: Authenticated-encryption with associated-data. In: Atluri, V. (ed.) Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002. pp. 98–107. ACM (2002), http://dl.acm.org/citation.cfm?id=586110
48. Saarinen, M.O.: Cycling attacks on gcm, GHASH and other polynomial macs and hashes. In: Canteaut, A. (ed.) FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers. Lecture Notes in Computer Science, vol. 7549, pp. 216–225. Springer (2012)
49. Shrimpton, T., Terashima, R.S.: Salvaging weak security bounds for blockcipher-based constructions. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10031, pp. 429–454 (2016)
50. Tao, B., Wu, H.: Improving the Biclique Cryptanalysis of AES. In: Foo, E., Stebila, D. (eds.) Information Security and Privacy - 20th Australasian Conference, ACISP 2015, Brisbane, QLD, Australia, June 29 - July 1, 2015, Proceedings. Lecture Notes in Computer Science, vol. 9144, pp. 39–56. Springer (2015)
51. Tessaro, S.: Optimally Secure Block Ciphers from Ideal Primitives. In: Iwata and Cheon [31], pp. 437–462
52. Wegman, M.N., Carter, L.: New Hash Functions and Their Use in Authentication and Set Equality. J. Comput. Syst. Sci. 22(3), 265–279 (1981)
53. Zaverucha, G.: Hybrid Encryption in the Multi-User Setting. Cryptology ePrint Archive, Report 2012/159 (2012), http://eprint.iacr.org/2012/159
54. Zhang, P., Hu, H.: On the provable security of the tweakable even-mansour cipher against multi-key and related-key attacks. Cryptology ePrint Archive, Report 2016/1172 (2016), http://eprint.iacr.org/2016/1172